

数值分析第三次上机练习实验报告

——插值法

力 4 杨昊光 2014011619

一、问题的描述

设 $f(x) = \frac{1}{1+25x^2}$, $x \in [-1, 1]$, 取 $x_j = -1 + \frac{2j}{n}$, $j = 0, 1, \dots, n$. 取适当的 n (比如 $n=10, 20, \dots$ 等), 试求出 n 次 Lagrange 插值多项式 $L_n(x)$ 、分段线性插值函数 $I_1^h(x)$ 和三次样条插值函数 $S_3^h(x)$ (采用自然边界条件), 画出它们的图像, 并对结果做一个比较说明。

二、方法描述

1. Lagrange 插值多项式的构造

设插值函数为 $L_n(x) = \sum_{j=0}^n y_j l_j(x)$, 其中 $y_j = f(x_j)$, l_j 为线性无关的 n 次多项式作为插值函数的基, 且满足:

$$l_j(x_m) = \delta_{jm} = \begin{cases} 1, & j = m \\ 0, & j \neq m \end{cases} \quad j, m = 0, 1, 2, \dots, n$$

故令 $l_j(x) = \prod_{i=0, i \neq j}^{j-1} \frac{x-x_i}{x_j-x_i} \cdot \prod_{i=j+1}^n \frac{x-x_i}{x_j-x_i} = \prod_{i=0, i \neq j}^n \frac{x-x_i}{x_j-x_i}$ $j = 0, 1, \dots, n$, 满足以上要求。

2. 分段线性插值函数的构造

类似地, 构造插值函数使之具有结构 $I_1^h(x) = \sum_{j=0}^n y_j \phi_j(x)$, 其中基函数 $\phi_n(x)$ 满足在相应的 x_n 处取 1, 在其余 x_j 处取 0, 并将各函数点线性连接。故作:

$$\phi_j(x) = \begin{cases} 0, & x \leq x_{j-1} \text{ 或 } x \geq x_{j+1}; \\ \frac{x-x_{j-1}}{x_j-x_{j-1}}, & x_{j-1} \leq x \leq x_j; \\ \frac{x-x_{j+1}}{x_j-x_{j+1}}, & x_j \leq x \leq x_{j+1}; \end{cases} \quad -1 \leq x \leq 1, j = 0, 1, 2, \dots, n$$

为分段线性插值的基函数。

3. 三次样条插值函数的构造

目的为令插值函数 $S_3^h(x) = \sum_{j=0}^n [f_j \alpha_j(x) + m_j \beta_j(x)]$ 在 x_j 处的函数值和一阶、二阶导数值连续且分别等于 $f_j(x_j)$ 、 $f_j'(x_j)$ 和 $f_j''(x_j)$ 。

若 $f_j'(x_j)$ 可知, 由分段 Hermite 插值可得函数形式 $S_3^h(x) = \sum_{j=0}^n [f_j \alpha_j(x) + m_j \beta_j(x)]$, 其中, 基函数为:

$$\alpha_j(x) = \begin{cases} \left(1 + \frac{2(x-x_j)}{x_{j-1}-x_j}\right) \left(\frac{x-x_{j-1}}{x_j-x_{j-1}}\right)^2, & x_{j-1} \leq x \leq x_j; \\ \left(1 + \frac{2(x-x_j)}{x_{j+1}-x_j}\right) \left(\frac{x-x_{j+1}}{x_j-x_{j+1}}\right)^2, & x_j \leq x \leq x_{j+1}; \\ 0, & x \leq x_{j-1} \text{ 或 } x \geq x_{j+1}; \end{cases} \quad -1 \leq x \leq 1, j = 0, 1, 2, \dots, n.$$

$$\beta_j(x) = \begin{cases} (x-x_j)\left(\frac{x-x_{j-1}}{x_j-x_{j-1}}\right)^2, & x_{j-1} \leq x \leq x_j; \\ (x-x_j)\left(\frac{x-x_{j+1}}{x_j-x_{j+1}}\right)^2, & x_j \leq x \leq x_{j+1}; \\ 0, & x \leq x_{j-1} \text{ 或 } x \geq x_{j+1}; \end{cases} \quad \begin{matrix} -1 \leq x \leq 1, \\ j = 0, 1, 2, \dots, n. \end{matrix}$$

系数为 f_j 和 m_j , 其中 m_j 满足:

$$\begin{aligned} \lambda_j m_{j-1} + 2m_j + \mu_j m_{j+1} &= g_j; \\ g_j &= 3 \left(\frac{\lambda_j (f(x_j) - f(x_{j-1}))}{x_j - x_{j-1}} + \frac{\mu_j (f(x_j) - f(x_{j+1}))}{x_j - x_{j+1}} \right), \quad j = 1, 2, \dots, n-1; \\ \lambda_j &= \frac{x_{j+1} - x_j}{x_{j+1} - x_{j-1}} = \frac{1}{2}, \quad \mu_j = 1 - \lambda_j = \frac{1}{2}. \end{aligned}$$

代入 g_j 表达式化简得:

$$g_j = \frac{3}{n} (f(x_{j+1}) - f(x_{j-1})), \quad j = 1, 2, \dots, n-1.$$

为了使计算方便, 这里取了第一类边界条件的转角方程, 即除了给出各 x_j 点的函数值之外, 还有: $f'(-1) = m_0 = -f'(1) = -m_n = -\frac{50x}{(1+25x^2)^2} \Big|_{x=-1} = \frac{25}{338}$

$$\text{故有: } \begin{pmatrix} 2 & \frac{1}{2} & & \\ \frac{1}{2} & 2 & \ddots & \\ & \ddots & \ddots & \frac{1}{2} \\ & & \frac{1}{2} & 2 \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_{n-2} \\ m_{n-1} \end{pmatrix} = \begin{pmatrix} g_1 - \frac{25}{676} \\ g_2 \\ \vdots \\ g_{n-2} \\ g_{n-1} + \frac{25}{676} \end{pmatrix}, \text{ 解出各 } m_j \text{ 后再代入算出插值函数。}$$

三、 方案设计

编写 Matlab 程序实现以上函数的构造和求值画出图像。程序包括如下几个部分, 最后再将其整体置于 n 从 5 隔 5 取到 100 的循环中以取得不同阶数的插值效果

- x_j 及 $f(x_j)$ 的计算: 作分划点 $x_j = -1 + \frac{2j}{n}$, $j = 0, 1, \dots, n$ 及在相应点的函数参照值 fref , 保存到两个数组中;
- Lagrange 插值: 先作出矩阵 $\left(\frac{x-x_i}{x_j-x_i}\right)_{j \times i = n \times n}$, 再按行连乘第 j 行除对角线之外的元素, 得到 $l_j(x)$ 构成的数组。最后将 $l_j(x)$ 数组与 fref 数组作向量内积, 并代入 x (从 -1 取到 1, 步长为 0.001) 的值, 得到 Lagrange 插值函数在 $x \in [-1, 1]$ 的函数值表并据此画出图像。将各点值与直接将 x 代入 $\frac{1}{1+25x^2}$ 所得的值相减得到误差并画图。
- 分段线性差值函数: 按定义构造基函数 $\phi_j(x)$, 同样列成数组与 fref 作内积并代入 x 的值, 画出函数图像。用与上面相同的方法进行误差分析。

- 三次样条插值函数: 按定义构造基函数 $\alpha_j(x)$ 和 $\beta_j(x)$, 以及不同 n 下的 $\begin{pmatrix} g_1 - \frac{25}{676} \\ g_2 \\ \vdots \\ g_{n-2} \\ g_{n-1} + \frac{25}{676} \end{pmatrix}$ 。

解出：

$$M = \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_{n-2} \\ m_{n-1} \end{pmatrix} = \begin{pmatrix} 2 & \frac{1}{2} & & \\ & \frac{1}{2} & 2 & \\ & & \ddots & \frac{1}{2} \\ & & & \frac{1}{2} & 2 \end{pmatrix}^{-1} \begin{pmatrix} g_1 - \frac{25}{676} \\ g_2 \\ \vdots \\ g_{n-2} \\ g_{n-1} + \frac{25}{676} \end{pmatrix} \text{ 以及 } m_0 = \frac{25}{338}, m_n = -\frac{25}{338},$$

作 fref 与 $(\alpha_j(x))$ 的内积和 M 与 $(\beta_j(x))$ 的内积之和，代入 x 的值，即为 $S_3^h(x)$ 。

同样按之前的方法作误差分析。

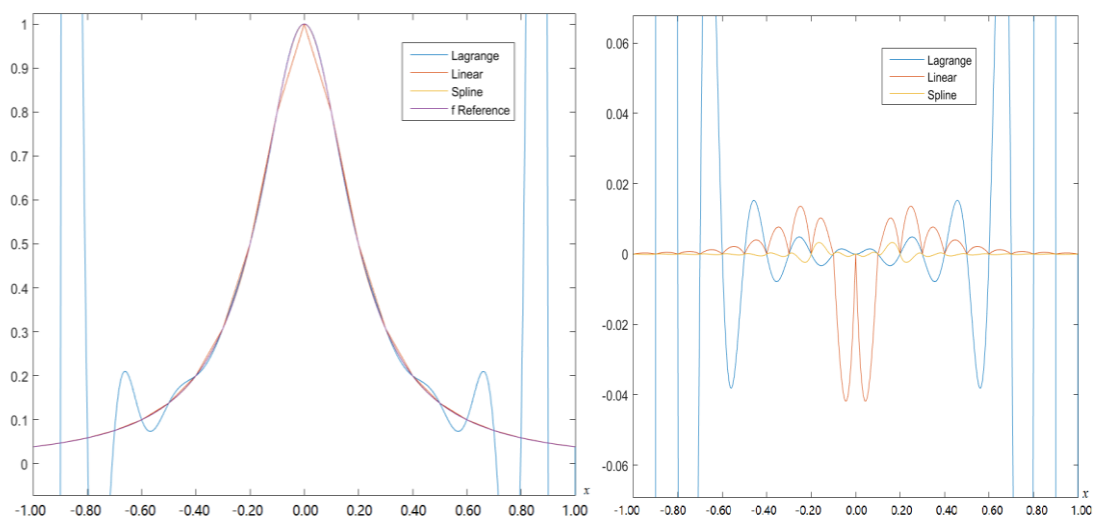
四、 计算结果及其分析

本次实验共取了 $n=5, 10, 15, \dots, 100$ 作为分析对象。作为参照的直接代入法求解的 $f(x)$ ，其 x 取值步长为 0.001，即一共取了 2000 个点进行函数误差值比较和画图。特别地， $n=20$ 时，各方法函数值和误差作列表如下：

x	$L_{20}(x)$	绝对误差	相对误差	$L_{20}(x)$	绝对误差	相对误差	$S_{20}(x)$	绝对误差	相对误差
-1.00	0.038462	0.000000	0.000000	0.038462	0.000000	0.000000	0.038462	0.000000	0.000000
-0.99	-42.470508	-42.509720	-1084.1041	0.039321	0.000109	0.002791	0.039206	-0.000006	-0.000140
-0.98	-58.238141	-58.278125	-1457.5359	0.040181	0.000197	0.004927	0.039964	-0.000020	-0.000488
-0.97	-58.544708	-58.585487	-1436.6626	0.041041	0.000262	0.006421	0.040741	-0.000038	-0.000940
-0.96	-50.864415	-50.906013	-1223.7805	0.041900	0.000303	0.007287	0.041539	-0.000058	-0.001400
-0.95	-39.952449	-39.994889	-942.37958	0.042760	0.000320	0.007537	0.042365	-0.000076	-0.001781
-0.94	-28.662567	-28.705875	-662.81866	0.043620	0.000311	0.007184	0.043222	-0.000087	-0.002004
-0.93	-18.553922	-18.598126	-420.73610	0.044480	0.000276	0.006241	0.044115	-0.000088	-0.001998
-0.92	-10.334508	-10.379634	-230.01270	0.045339	0.000213	0.004720	0.045050	-0.000077	-0.001698
-0.91	-4.179237	-4.225315	-91.699896	0.046199	0.000121	0.002636	0.046029	-0.000048	-0.001049
-0.90	0.047059	0.000000	0.000000	0.047059	0.000000	0.000000	0.047059	0.000000	0.000000
-0.89	2.637774	2.589703	53.872288	0.048235	0.000164	0.003415	0.048113	0.000042	0.000872
-0.88	3.945073	3.895957	79.321682	0.049412	0.000296	0.006024	0.049170	0.000054	0.001099
-0.87	4.318304	4.268110	85.031420	0.050588	0.000394	0.007844	0.050239	0.000044	0.000878
-0.86	4.069132	4.017824	78.307386	0.051765	0.000456	0.008894	0.051328	0.000020	0.000388
-0.85	3.454958	3.402499	64.860133	0.052941	0.000482	0.009191	0.052448	-0.000011	-0.000207
-0.84	2.674355	2.620707	48.849983	0.054118	0.000470	0.008753	0.053608	-0.000041	-0.000755
-0.83	1.869927	1.815050	33.074747	0.055294	0.000417	0.007597	0.054816	-0.000062	-0.001124
-0.82	1.135307	1.079158	19.219811	0.056471	0.000322	0.005741	0.056081	-0.000067	-0.001190
-0.81	0.524038	0.466575	8.119564	0.057647	0.000184	0.003203	0.057414	-0.000049	-0.000847
-0.80	0.058824	0.000000	0.000000	0.058824	0.000000	0.000000	0.058824	0.000000	0.000000
-0.79	-0.259795	-0.320027	-5.313247	0.060488	0.000256	0.004258	0.060276	0.000044	0.000735
-0.78	-0.445868	-0.507558	-8.227515	0.062153	0.000463	0.007503	0.061742	0.000052	0.000835
-0.77	-0.521807	-0.585008	-9.256294	0.063818	0.000617	0.009760	0.063233	0.000032	0.000510
-0.76	-0.513554	-0.578321	-8.929274	0.065483	0.000716	0.011054	0.064763	-0.000003	-0.000052
-0.75	-0.447052	-0.513442	-7.733720	0.067148	0.000758	0.011411	0.066345	-0.000045	-0.000678
-0.70	0.075472	0.000000	0.000000	0.075472	0.000000	0.000000	0.075472	0.000000	0.000000
-0.60	0.100000	0.000000	0.000000	0.100000	0.000000	0.000000	0.100000	0.000000	0.000000
-0.50	0.137931	0.000000	0.000000	0.137931	0.000000	0.000000	0.137931	0.000000	0.000000
-0.40	0.200000	0.000000	0.000000	0.200000	0.000000	0.000000	0.200000	0.000000	0.000000
-0.30	0.307692	0.000000	0.000000	0.307692	0.000000	0.000000	0.307692	0.000000	0.000000
-0.20	0.500000	0.000000	0.000000	0.500000	0.000000	0.000000	0.500000	0.000000	0.000000
-0.10	0.800000	0.000000	0.000000	0.800000	0.000000	0.000000	0.800000	0.000000	0.000000
0.00	1.000000	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000	0.000000	0.000000
0.10	0.800000	0.000000	0.000000	0.800000	0.000000	0.000000	0.800000	0.000000	0.000000
0.20	0.500000	0.000000	0.000000	0.500000	0.000000	0.000000	0.500000	0.000000	0.000000
0.30	0.307692	0.000000	0.000000	0.307692	0.000000	0.000000	0.307692	0.000000	0.000000
0.40	0.200000	0.000000	0.000000	0.200000	0.000000	0.000000	0.200000	0.000000	0.000000
0.50	0.137931	0.000000	0.000000	0.137931	0.000000	0.000000	0.137931	0.000000	0.000000
0.60	0.100000	0.000000	0.000000	0.100000	0.000000	0.000000	0.100000	0.000000	0.000000
0.70	0.075472	0.000000	0.000000	0.075472	0.000000	0.000000	0.075472	0.000000	0.000000
0.75	-0.447052	-0.513442	-7.733720	0.067148	0.000758	0.011411	0.066345	-0.000045	-0.000678
0.76	-0.513554	-0.578321	-8.929274	0.065483	0.000716	0.011054	0.064763	-0.000003	-0.000052
0.77	-0.521807	-0.585008	-9.256294	0.063818	0.000617	0.009760	0.063233	0.000032	0.000510
0.78	-0.445868	-0.507558	-8.227515	0.062153	0.000463	0.007503	0.061742	0.000052	0.000835
0.79	-0.259795	-0.320027	-5.313247	0.060488	0.000256	0.004258	0.060276	0.000044	0.000735
0.80	0.058824	0.000000	0.000000	0.058824	0.000000	0.000000	0.058824	0.000000	0.000000

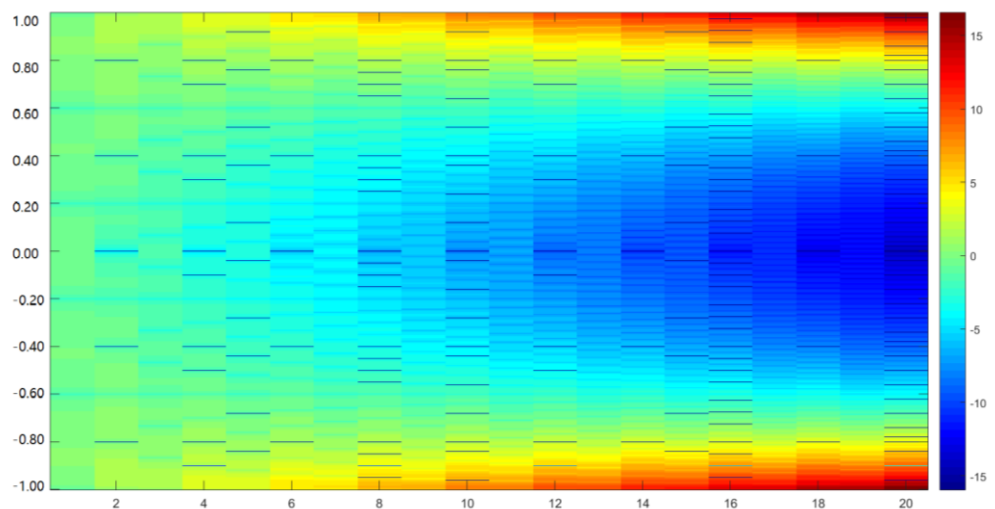
0.81	0.524038	0.466575	8.119564	0.057647	0.000184	0.003203	0.057414	-0.000049	-0.000847
0.82	1.135307	1.079158	19.219811	0.056471	0.000322	0.005741	0.056081	-0.000067	-0.001190
0.83	1.869927	1.815050	33.074747	0.055294	0.000417	0.007597	0.054816	-0.000062	-0.001124
0.84	2.674355	2.620707	48.849983	0.054118	0.000470	0.008753	0.053608	-0.000041	-0.000755
0.85	3.454958	3.402499	64.860133	0.052941	0.000482	0.009191	0.052448	-0.000011	-0.000207
0.86	4.069132	4.017824	78.307386	0.051765	0.000456	0.008894	0.051328	0.000020	0.000388
0.87	4.318304	4.268110	85.031420	0.050588	0.000394	0.007844	0.050239	0.000044	0.000878
0.88	3.945073	3.895957	79.321682	0.049412	0.000296	0.006024	0.049170	0.000054	0.001099
0.89	2.637774	2.589703	53.872288	0.048235	0.000164	0.003415	0.048113	0.000042	0.000872
0.90	0.047059	0.000000	0.000000	0.047059	0.000000	0.000000	0.047059	0.000000	0.000000
0.91	-4.179237	-4.225315	-91.699896	0.046199	0.000121	0.002636	0.046029	-0.000048	-0.001049
0.92	-10.334508	-10.379634	-230.01270	0.045339	0.000213	0.004720	0.045050	-0.000077	-0.001698
0.93	-18.553922	-18.598126	-420.73610	0.044480	0.000276	0.006241	0.044115	-0.000088	-0.001998
0.94	-28.662567	-28.705875	-662.81866	0.043620	0.000311	0.007184	0.043222	-0.000087	-0.002004
0.95	-39.952449	-39.994889	-942.37958	0.042760	0.000320	0.007537	0.042365	-0.000076	-0.001781
0.96	-50.864415	-50.906013	-1223.7805	0.041900	0.000303	0.007287	0.041539	-0.000058	-0.001400
0.97	-58.544708	-58.585487	-1436.6626	0.041041	0.000262	0.006421	0.040741	-0.000038	-0.000940
0.98	-58.238141	-58.278125	-1457.5359	0.040181	0.000197	0.004927	0.039964	-0.000020	-0.000488
0.99	-42.470508	-42.509720	-1084.1041	0.039321	0.000109	0.002791	0.039206	-0.000006	-0.000140
1.00	0.038462	0.000000	0.000000	0.038462	0.000000	0.000000	0.038462	0.000000	0.000000

各法插值函数及误差如下图所示（左为插值函数图，右为误差图）：



各种方法在不同阶数下的相对误差如下图所示，横坐标为阶数，纵坐标为 x ，为了便于演示，图像中的值对应取相对误差的绝对值对数： $z = \log\left(\left|\frac{p_n(x)-f(x)}{f(x)}\right|\right)$ 。

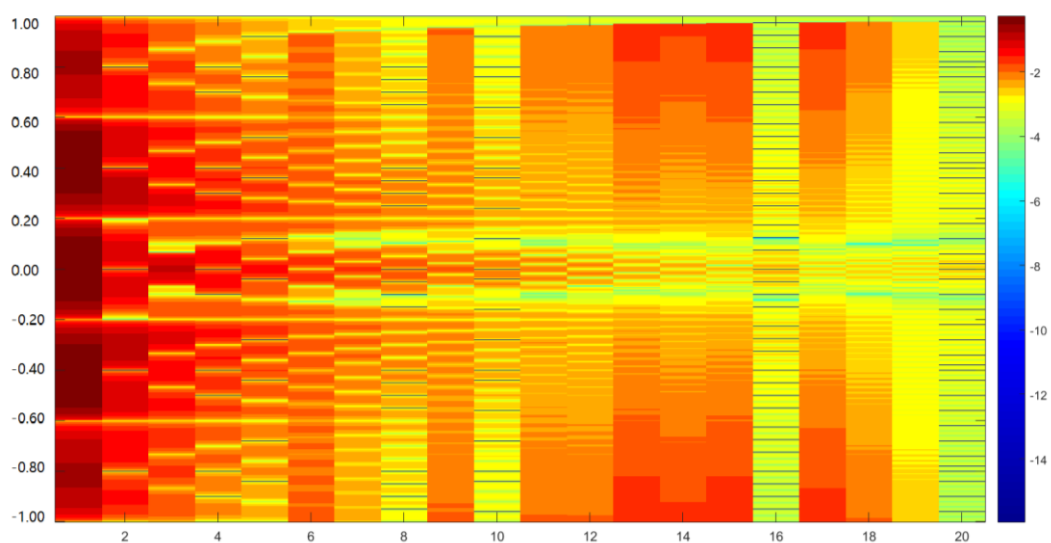
Lagrange 方法：



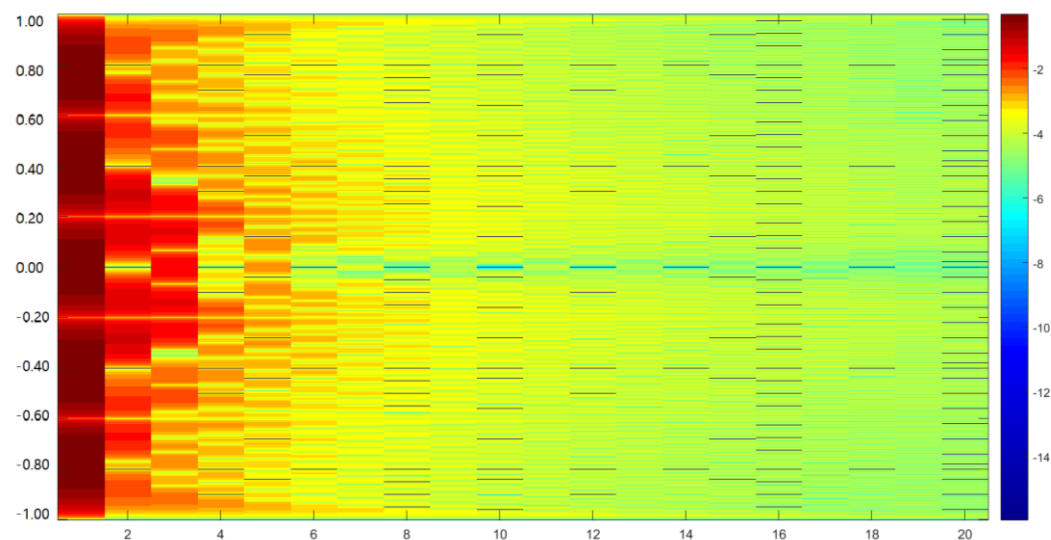
由图可见，Lagrange 插值法在边缘有很严重的 Runge 现象，而且阶数越高，虽然在插值区间中部误差减小，然而在边缘出现同样误差 Runge 现象时 $|x|$ 越小；然而阶数较低

时在整个区间上精度都不高。

分段线性插值：



三次样条插值：



对比可知，后两者在高阶时的表现都比较稳定，三次样条插值法比分段线性插值法更胜一筹。然而，对于局部的插值精度，比如说 $[-0.50, 0.50]$ 上的插值精度，高阶 Lagrange 插值法在高阶时比后两者更好一些。

五、 结论

多项式插值是最为常见的函数插值方法。函数插值方法目的在于找到一个函数的近似，使其在给定的点上等于给定值或导数的给定值，并在给定点上具有尽可能好的性质，如连续性、一阶和二阶导数的连续性。

常见的多项式插值包括 Lagrange 插值、Newton 插值、Hermite 插值、分段线性插值、分段 Hermite 插值、样条插值等。对于整体的插值法，理论上插值函数阶数越高，越接近目标函数，然而在边界上可能会因为系数之和收敛而导致病态的发散现象即 Runge 现象。在本次实验中，这种现象的确出现了。为了确保整体上的精度，我们可以转而选用分段插值的方法如分段线性插值法和样条插值法。其中，分段样条插值因其函

数性质较好,可以满足给定点上函数值和一、二阶导数值的连续性,因而整体精度也比线性插值高。然而有趣的是,当阶数足够高时 **Lagrange** 法等整体的多项式插值在区间内部的插值精度比同等阶数的分段插值的要高,可能是因为多项式插值的函数性质比分段函数的性质要更好一些的缘故。